

画像は 16×16 個の世界である：における画像認識のため のトランスフォーマー

CALE

**Alexey Dosovitskiy^{*,†}, Lucas Beyer^{*}, Alexander Kolesnikov^{*}, Dirk Weissenborn^{*},
Xiaohua Zhai^{*}, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby^{*,†}**

^{*}equal technical contribution, [†]equal advising

Google Research, Brain Team

{adosovitskiy, neilhoulsby}@google.com

ABSTRACT

Transformerアーキテクチャは自然言語処理タスクのデファクトスタンダードとなっているが、コンピュータビジョンへの応用はまだ限定的である。視覚では、注意は畳み込みネットワークと組み合わせて適用されるか、畳み込みネットワークの全体的な構造を維持したまま、特定のコンポーネントを置き換えるために使用される。このようなCNNへの依存は必要なく、画像パッチのシーケンスに直接適用される純粋な変換器は、画像分類タスクで非常に良い性能を発揮できることを示す。大量のデータで事前学習し、複数の中規模または小規模の画像認識ベンチマーク(ImageNet、CIFAR-100、VTABなど)に転送すると、Vision Transformer(ViT)は、学習に必要な計算資源を大幅に削減しながら、最先端の畳み込みネットワークと比較して優れた結果を達成する¹。

1 INTRODUCTION

自己注意に基づくアーキテクチャ、特にTransformers (Vaswani et al., 2017)は、自然言語処理(NLP)で選択されるモデルとなっている。主要なアプローチは、大規模なテキストコーパスで事前学習し、その後、より小さなタスク固有のデータセットで微調整を行うことである(Devlin et al.)Transformersの計算効率とスケーラビリティのおかげで、100B以上のパラメータを持つ、前例のないサイズのモデルを訓練することが可能になった(Brown et al.)モデルやデータセットが大きくなても、まだ性能が飽和する気配はない。

しかし、コンピュータビジョンでは、畳み込みアーキテクチャが依然として支配的である(LeCun et al., 1989; Krizhevsky et al., 2012; He et al., 2016)。NLPの成功に触発され、複数の作品がCNNのようなアーキテクチャと自己注意を組み合わせようとし(Wangら、2018;Carionら、2020)、一部は畳み込みを完全に置き換える(Ramachandranら、2019;Wangら、2020a)。後者のモデルは、理論的には効率的であるが、特殊な注意パターンを使用するため、最新のハードウェアアクセラレータではまだ効果的にスケールされていない。したがって、大規模な画像認識において、古典的なResNetライクなアーキテクチャは依然として最先端である(Mahajan et al., 2018; Xie et al., 2020; Kolesnikov et al., 2020)。

NLPにおけるTransformerのスケーリングの成功に触発され、我々は標準的なTransformerを画像に直接適用する実験を行ったが、可能な修正は最も少なかった。そのために、画像をパッチに分割し、これらのパッチの線形埋め込みシーケンスをTransformerの入力として提供する。画像パッチは、自然言語処理アプリケーションのトークン(単語)と同じように扱われる。教師あり方式で画像分類のモデルを学習する。

ImageNetのような中規模のデータセットで強力な正則化なしで学習した場合、これらのモデルは同等のサイズのResNetsより数%ポイント低い控えめな精度をもたらす。この一見落胆するような結果は予想通りかもしれない：トランスフォーマーは、

¹Fine-tuning code and pre-trained models are available at https://github.com/google-research/vision_transformer

並進の等価性や局所性など、CNNに固有の帰納的バイアスをいくつか欠いているため、不十分なデータ量で訓練した場合にはうまく汎化できない。

しかし、より大きなデータセット(14M–300M画像)でモデルを学習させると、様相は変化する。大規模なトレーニングは帰納的バイアスに勝ることがわかった。我々のVision Transformer (ViT)は、十分なスケールで事前学習され、より少ないデータポイントのタスクに転送された場合、優れた結果を達成する。公開されているImageNet-21kデータセットや社内のJFT-300Mデータセットで事前学習した場合、ViTは複数の画像認識ベンチマークで最先端技術に近づくか、それを上回る。特に、ImageNetで88.55%、ImageNet-Realで90.72%、CIFAR-100で94.55%、VTABの19タスク群で77.63%の精度を達成した。

2 RELATED WORK

Transformerは機械翻訳のためにVaswaniら(2017)によって提案され、それ以来多くのNLPタスクにおける最先端の手法となっている。大規模なTransformerベースのモデルは、多くの場合、大規模なコーパスで事前に訓練され、その後、手元のタスクのために微調整される：BERT (Devlin et al., 2019)はノイズ除去の自己教師付き事前学習タスクを使用し、GPTのラインは事前学習タスクとして言語モデリングを使用する(Radford et al., 2018; 2019; Brown et al., 2020)。

自己アテンションを画像にナープに適用すると、各画素が他の全ての画素にアテンションする必要がある。ピクセル数の2次コストでは、現実的な入力サイズには対応できない。このように、画像処理の文脈でTransformerを適用するために、過去にいくつかの近似が試みられてきた。Parmarら(2018)は、グローバルではなく、各クエリピクセルのローカル近傍でのみ自己注意を適用した。このような局所的な多頭ドット積自己注意ブロックは、畳み込みを完全に置き換えることができる(Hu et al., 2019; Ramachandran et al., 2019; Zhao et al., 2020)。別の研究では、Sparse Transformers (Child et al., 2019)は、画像に適用できるように、大域的自己注意のスケーラブルな近似を採用している。注意をスケールする別の方針は、様々なサイズのブロックで注意を適用することであり(Weissenborn et al., 2019)、極端な場合には個々の軸に沿ってのみ注意を適用する(Ho et al., 2019; Wang et al., 2020a)。これらの特殊な注意アーキテクチャの多くは、コンピュータビジョンタスクで有望な結果を示すが、ハードウェアアクセラレータで効率的に実装するためには、複雑なエンジニアリングが必要である。

我々のモデルと最も関連するのは、Cordonnierら(2020)のモデルで、入力画像からサイズ 2×2 のパッチを抽出し、その上に完全な自己注意を適用するものである。このモデルはViTと非常によく似ているが、我々の研究はさらに進んで、大規模な事前学習により、バニラ変換器が最先端のCNNと遜色ない(あるいはそれ以上の)ことを実証している。さらに、Cordonnierら(2020)は 2×2 ピクセルの小さなパッチサイズを用いているため、モデルは小さな解像度の画像にのみ適用可能であるが、我々は中解像度の画像も扱う。

また、畳み込みニューラルネットワーク(CNN)と自己注意の形態を組み合わせること、例えば画像分類のための特徴マップを増強すること(Bello et al., 2019)、あるいは自己注意を用いてCNNの出力をさらに処理すること、例えば物体検出(Hu et al., 2018; Carion et al., 2020)、ビデオ処理(Wang et al., 2018; Sun et al., 2019)、画像分類(Wu et al., 2020)、教師なし物体発見(Locatello et al., 2020)、または統一テキスト視覚タスク(Chenら, 2020c; Luら, 2019; Liら, 2019)。

もう一つの最近の関連モデルは、画像GPT(iGPT)(Chen et al., 2020a)であり、画像の解像度と色空間を縮小した後に画像ピクセルにTransformerを適用する。このモデルは生成モデルとして教師無しで学習され、得られた表現は分類性能のために線形に微調整またはプローブされ、ImageNetで最大72%の精度を達成することができる。

我々の研究は、標準的なImageNetデータセットよりも大規模な画像認識を探求する論文の増加コレクションに追加するものである。追加データソースを使用することで、標準的なベンチマークで最先端の結果を達成することができる(Mahajan et al., 2018; Touvron et al., 2019; Xie et al., 2020)。さらに、Sunら(2017)はCNNの性能がデータセットサイズによってどのようにスケールするかを研究し、Kolesnikovら(2020); Djolongaら(2020)はImageNet-21kやJFT-300Mのような大規模データセットからCNN転移学習の実証的な探索を行う。我々はこれら2つの後者のデータセットにも注目するが、先行研究で用いられたResNetベースのモデルの代わりにTransformerを訓練する。

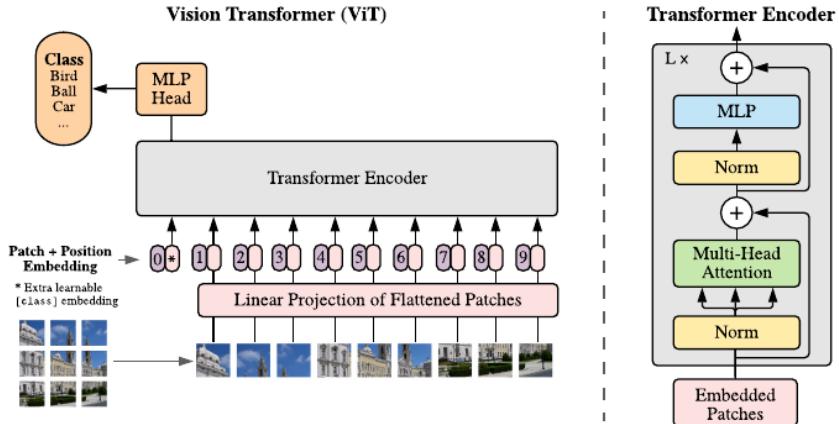


図1:モデルの概要。画像を固定サイズのパッチに分割し、それぞれを線形に埋め込み、位置埋め込みを追加し、得られたベクトル列を標準的なTransformerエンコーダに与える。分類を行うために、学習可能な「分類トークン」をシーケンスに追加するという標準的なアプローチを用いる。Transformerエンコーダの図は、Vaswaniら(2017)に触発されたものである。

3 METHOD

モデル設計では、オリジナルのTransformer(Vaswani et al.)この意図的に単純なセットアップの利点は、スケーラブルなNLP Transformerアーキテクチャとその効率的な実装が、ほぼそのまま使えることである。

3.1 ビジョン変換器(VIT)

モデルの概要を図1に示す。標準的なTransformerはトークン埋込みの1次元シーケンスを入力として受け取る。2次元画像を扱うために、画像 $x \in \mathbb{R}^{H \times W \times C}$ を平坦化された2次元パッチの列 $x_p \in \mathbb{R}^{N \times (P^2 - C)}$ に再形成する、Cはチャンネル数、(P, P)は各画像パッチの解像度、N = HW/P² は結果のパッチ数であり、これはTransformerの有効入力シーケンス長としても機能する。Transformerは全ての層を通して一定の潜在ベクトルサイズDを使うので、パッチを平坦化し、学習可能な線形射影(式1)でD次元に写像する。この射影の出力をパッチ埋め込みと呼ぶことにする。

BERTの[class]トークンと同様に、埋め込みパッチ列($z_0 = x_{class}$)に学習可能な埋め込みを付加し、そのTransformerエンコーダの出力における状態(z_L^0)が画像表現yとなる(式4)。事前学習時と微調整時の両方で、 z_L^0 に分類ヘッドが取り付けられる。分類ヘッドは、事前学習時には1つの隠れ層を持つMLPによって実装され、微調整時には1つの線形層によって実装される。

位置情報を保持するために、パッチ埋め込みに位置埋め込みを追加する。より高度な2D認識位置埋め込み(付録D.4)を用いても、大きな性能向上は観察されなかったので、標準的な学習可能な1D位置埋め込みを使用する。結果として得られる埋め込みベクトル列はエンコーダへの入力となる。

Transformerエンコーダ(Vaswani et al., 2017)は、多頭自己注意(MSA、付録A参照)とMLPブロック(式2、3)の交互の層で構成される。レイヤーノーム(LN)は各ブロックの前に適用され、残差接続は各ブロックの後に適用される(Wang et al.)

MLPはGELU非線形性を持つ2つの層を含む。

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}}, \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D} \quad (1)$$

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \dots L \quad (2)$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \quad \ell = 1 \dots L \quad (3)$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0) \quad (4)$$

帰納的バイアス。Vision TransformerはCNNよりも画像固有の帰納的バイアスがはるかに少ないと注意。CNNでは、局所性、2次元近傍構造、並進等価性がモデル全体を通して各層に焼き付けられる。ViTでは、MLP層のみが局所的で並進等変であり、自己注意層はグローバルである。2次元近傍構造は、非常に控えめに使用される：モデルの初期には、画像をパッチに切断し、微調整の時点で、異なる解像度の画像（後述）の位置埋め込みを調整する。それ以外には、初期化時の位置埋め込みはパッチの2次元位置に関する情報を持たず、パッチ間の全ての空間関係をゼロから学習する必要がある。

ハイブリッドアーキテクチャ。生の画像パッチの代替として、入力シーケンスはCNNの特徴マップから形成することができる（LeCun et al., 1989）。このハイブリッドモデルでは、CNN特徴マップから抽出されたパッチにパッチ埋め込み射影E式(1)を適用する。特殊なケースとして、パッチは空間サイズ1x1を持つことができ、これは入力シーケンスが単に特徴マップの空間次元を平坦化し、Transformer次元に射影することによって得られることを意味する。分類入力の埋め込みと位置の埋め込みは、上記のように追加される。

3.2 チューニングと高分解能化

通常、ViTは大規模なデータセットで事前学習し、（小規模な）下流タスクに微調整を行う。このために、事前に学習された予測ヘッドを削除し、ゼロ初期化されたD×Kフィードフォワード層を追加する。事前学習よりも高い解像度で微調整を行うことが有益な場合が多い（Touvron et al., 2019; Kolesnikov et al., 2020）。より高解像度の画像を与える場合、パッチサイズは同じに保たれ、その結果、有効シーケンス長が大きくなる。Vision Transformerは（メモリ制約まで）任意のシーケンス長を扱うことができるが、事前に訓練された位置埋め込みはもはや意味をなさないかもしれない。そこで、事前に学習した位置埋め込みを、元画像中の位置に応じて2次元補間する。この解像度調整とパッチ抽出は、画像の2D構造に関する帰納的バイアスがVision Transformerに手動で注入される唯一のポイントであることに注意。

4 EXPERIMENTS

ResNet、Vision Transformer(ViT)、およびハイブリッドの表現学習能力を評価する。各モデルのデータ要件を理解するために、様々なサイズのデータセットで事前学習を行い、多くのベンチマークタスクを評価する。モデルの事前学習にかかる計算コストを考慮すると、ViTは非常に有利な性能を示し、より低い事前学習コストで、ほとんどの認識ベンチマークで最先端技術を達成する。最後に、自己監視を用いた小規模な実験を行い、自己監視付きViTが将来的に有望であることを示す。

4.1 SETUP

データセット。モデルのスケーラビリティを探るために、1kクラス、1.3M画像（以下ではImageNetと呼ぶ）、21kクラス、14M画像（Deng et al., 2009）のスーパーセットImageNet-21k、18kクラス、303M高解像度画像を持つJFT（Sun et al., 2017）を用いたILSVRC-2012 ImageNetデータセットを用いる。Kolesnikovら（2020）に従い、下流タスクのテストセットに対して事前学習データセットを重複排除する。これらのデータセットで学習したモデルを、いくつかのベンチマークタスクに移植する：オリジナルの検証ラベルとクリーンアップされたRealラベルのImageNet（Beyer et al., 2020）、CIFAR-10/100（Krizhevsky, 2009）、Oxford-IIIT Pets（Parkhi et al., 2012）、Oxford Flowers-102（Nilsback & Zisserman, 2008）。これらのデータセットについては、Kolesnikov et al.

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

表1: Vision Transformerモデルバリエントの詳細。

また、19タスクのVTAB分類スイート(Zhai et al., 2019) VTABは、タスクごとに1,000の学習例を用いて、多様なタスクへの低データ転送を評価する。タスクは3つのグループに分けられる：自然 - 上記のようなタスク、ペット、CIFARなど。特殊なタスク - 医療や衛星画像、構造化されたタスク - ローカライゼーションのような幾何的な理解を必要とするタスク。

モデルのバリエーション。表 1 に要約されているように、BERT(Devlin et al., 2019)に使用されるものをベースに ViT 構成を行う。ベース」と「大規模」モデルは、BERT から直接採用し、より大きな「Huge」モデルを追加する。以下では、モデルサイズと入力パッチサイズを示すために簡単な表記を用いる。例えば、ViT-L/16は、入力パッチサイズが 16×16 の「Large」変種を意味する。Transformerのシーケンス長はパッチサイズの2乗に反比例するので、パッチサイズが小さいモデルは計算コストが高くなることに注意してください。

ベースラインCNNについては、ResNet (He et al., 2016)を使用するが、パッチ正規化層(Ioffe & Szegedy, 2015)をグループ正規化(Wu & He, 2018)に置き換え、標準化畳み込み(Qiao et al., 2019)を使用する。これらの修正により転送が改善され(Kolesnikov et al., 2020)、修正モデルを「ResNet (BiT)」と呼ぶことにする。ハイブリッドでは、中間特徴マップを1つの”ピクセル”のパッチサイズでViTに入力する。異なるシーケンス長を実験するために、(i)通常のResNet50のステージ4の出力を取るか、(ii)ステージ4を削除し、ステージ3に同じ数のレイヤーを配置し(レイヤーの総数を維持する)、この拡張ステージ3の出力を取るかのいずれかである。オプション(ii)は、シーケンス長を4倍長くし、より高価なViTモデルになる。

学習と微調整。Adam (Kingma & Ba, 2015)を用いて、 $\beta_1 = 0.9$, $\beta_2 = 0.999$, パッチサイズ4096で、ResNetsを含む全てのモデルを訓練し、0.1という高い重み減衰を適用する。線形学習率のウォームアップと減衰を使用する。詳細は付録B.1を参照のこと。微調整には、すべてのモデルでモメンタム付きSGD、パッチサイズ512を使用した(付録B.1.1参照)。表2のImageNetの結果については、より高い解像度で微調整を行った：ViT-L/16では512、ViT-H/14では518であり、Polyak & Juditsky (1992)の平均化も0.9999倍で使用した(Ramachandran et al., 2019; Wang et al., 2020b)。

メトリクス。下流のデータセットについて、数ショットまたは微調整の精度で結果を報告する。微調整精度は、それぞれのデータセットで微調整を行った後の各モデルの性能を把握する。学習画像の部分集合の(凍結された)表現を $\{-1, 1\}^K$ ターゲットベクトルに写像する正則化最小二乗回帰問題を解くことにより、数ショット精度が得られる。この定式化により、厳密解を閉じた形で復元することができる。我々は主に微調整性能に焦点を当てているが、微調整がコスト高になりすぎるような高速なオンザフライ評価のために、線形数ショット精度を使用することもある。

4.2 C4.2 最先端技術との比較

まず、我々の最大のモデルであるViT-H/14とViT-L/16を、文献にある最先端のCNNと比較する。最初の比較点は、大きなResNetsで教師あり転送学習を行うBig Transfer (BiT) (Kolesnikov et al., 2020)である。2つ目はNoisy Student (Xie et al., 2020)で、これはImageNetとJFT300Mの半教師付き学習を用いてラベルを除去して学習した大規模なEfficientNetである。現在、ImageNetとBiT-Lでは、Noisy Studentが最先端である。すべてのモデルはTPUv3ハードウェアで学習され、それぞれの事前学習に要したTPUv3コア日数、つまり学習に使用したTPU v3コア数(1チップあたり2コア)に学習時間(日)を掛けたものを報告する。

表2に結果を示す。JFT-300Mで事前学習した小型のViT-L/16モデルは、全てのタスクにおいてBiT-L(同じデータセットで事前学習済み)を上回り、学習に必要な計算資源は大幅に削減された。より大きなモデルであるViT-H/14は、特にImageNet、CIFAR-100、VTABスイートといった難易度の高いデータセットにおいて、性能をさらに向上させる。

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	90.72 ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	99.50 ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	—
CIFAR-100	94.55 ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	—
Oxford-IIIT Pets	97.56 ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	—
Oxford Flowers-102	99.68 ± 0.02	99.74 ± 0.00	99.61 ± 0.02	99.63 ± 0.03	—
VTAB (19 tasks)	77.63 ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

表2:一般的な画像分類ベンチマークにおける最新技術との比較。3回の微調整を行った平均と標準偏差を報告する。JFT-300Mデータセットで事前学習したVision Transformerモデルは、すべてのデータセットでResNetベースのベースラインを上回り、事前学習に要する計算資源は大幅に削減された。より小さな公開ImageNet-21kデータセットで事前学習したViTも良好な結果を示した。* Touvron et al. (2020)で報告された88.5%の結果をわずかに改善した。

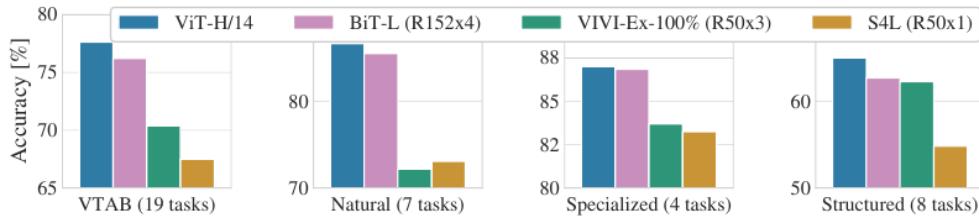


図2:Natural、Specialized、StructuredタスクグループにおけるVTAB性能の内訳。

興味深いことに、このモデルは事前学習にかかる計算量が先行技術より大幅に少ない。しかし、事前学習の効率はアーキテクチャの選択だけでなく、学習スケジュール、オプティマイザ、ウェイト減衰などの他のパラメータにも影響される可能性があることに注意する。セクション4.4では、異なるアーキテクチャの性能対計算量の対照研究を提供する。最後に、公開されているImageNet-21kデータセットで事前学習したViT-L/16モデルは、事前学習にかかるリソースが少なく、ほとんどのデータセットでも良好な性能を発揮する:8コアの標準クラウドTPUv3を用いて、約30日程度。

図2はVTABタスクをそれぞれのグループに分解し、このベンチマークにおける過去のSOTA手法と比較したものである: BiT、VIVI - ImageNetとYoutubeで協調学習したResNet(Tschannen et al., 2020)、S4L - ImageNetで教師あり+半教師あり学習(Zhai et al., 2019a)。ViT-H/14は、NaturalタスクとStructuredタスクにおいて、BiT-R152x4や他の手法を凌駕している。Specializedでは、上位2つのモデルの性能はほぼ同じである。

4.3 再トレーニングデータ要件

Vision Transformerは、大規模なJFT-300Mデータセットで事前学習した場合、良好な性能を発揮する。ResNetsよりも視覚的帰納的バイアスが少ないので、データセットサイズはどの程度重要か?我々は2つのシリーズの実験を行う。

まず、サイズが大きくなるデータセットでViTモデルを事前学習する: ImageNet、ImageNet-21k、JFT300M。より小さなデータセットでの性能向上させるために、3つの基本的な正則化パラメータ(重み減衰、ドロップアウト、ラベルスマージング)を最適化する。図3はImageNetにファインチューニングした後の結果である(他のデータセットでの結果は表5に示す)²。最小のデータセットで事前学習した場合、ImageNet、ViT-Largeモデルは、(中程度の)正則化にもかかわらず、ViT-Baseモデルと比較して性能が劣る。ImageNet-21kの事前学習では、両者の性能はほぼ同じである。JFT-300Mでのみ、より大きなモデルの利点が十分に見られる。図3は、以下の性能も示している。

² ImageNet で事前学習したモデルも微調整されているが、再び ImageNet で微調整されていることに注意。これは、微調整の際に解像度を上げると性能が向上するためである。

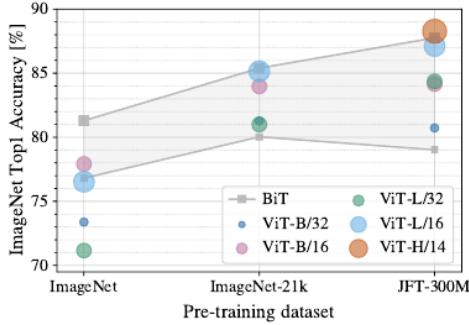


図3:ImageNetへの転送。大規模なViTモデルは、小規模なデータセットで事前学習した場合、BiT ResNets(斜線部分)よりも性能が劣るが、大規模なデータセットで事前学習した場合には、輝きを放つ。同様に、データセットが大きくなるにつれて、より大きなViTバリエントはより小さなバリエントを追い越す。

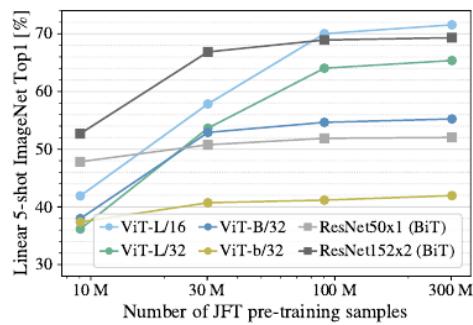


図4:ImageNetにおける事前学習サイズに対する線形少数ショット評価。ResNetsは事前学習データセットが小さいほど良い性能を示すが、事前学習が大きいほど良い性能を示すViTよりも早くプラトーする。ViT-bは、すべての隠れ次元を半分にしたViT-Bである。

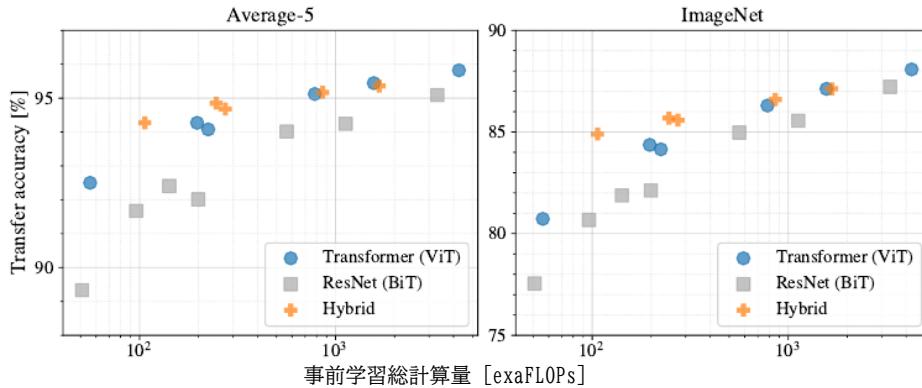


図5:異なるアーキテクチャの性能対事前学習計算量: Vision Transformers、ResNets、およびハイブリッド。Vision Transformersは、同じ計算予算でResNetsを概ね上回る。ハイブリッドは、モデルサイズが小さいほど純粋なTransformerより改善されるが、モデルサイズが大きくなるとその差はなくなる。

サイズの異なるBiTモデルによってスパンされる領域。BiT CNNはImageNetではViTを上回るが、より大きなデータセットではViTが上回る。

次に、9M、30M、90MのランダムなサブセットとJFT300Mの完全なデータセットでモデルを学習する。我々は、より小さな部分集合に対して追加の正則化を行わず、すべての設定に対して同じハイパーパラメータを使用する。このようにして、正則化の効果ではなく、モデル固有の特性を評価する。しかし、我々は早期停止を使用し、トレーニング中に達成された最高の検証精度を報告する。計算量を節約するため、完全な微調整精度ではなく、数ショットの線形精度を報告する。図4はその結果である。Vision TransformersはResNetsよりもオーバーフィットしており、より小さなデータセットでも同等の計算コストがかかる。例えば、ViT-B/32はResNet50より若干高速である。9Mサブセットでは性能が大幅に低下するが、90M以上のサブセットでは性能が向上する。ResNet152x2とViT-L/16についても同様である。この結果は、畳み込み誘導バイアスは小さいデータセットには有効であるが、大きいデータセットではデータから直接関連するパターンを学習すれば十分であり、有益できあえるという直観を補強するものである。

全体として、ImageNetの数ショットの結果(図4)と、VTABの低データの結果(表2)は、非常に低データ転送に有望であると思われる。ViTの数ショット特性のさらなる解析は、今後の研究のエキサイティングな方向性である。

4.4 スケーリング研究

JFT-300Mからの転送性能を評価することで、異なるモデルの制御されたスケーリング研究を行う。この設定では、データサイズはモデルの性能をボトルネックにせず、各モデルの性能対事前学習コストを評価する。モデルセットは以下の通りである：7つのResNets、R50x1、R50x2 R 101x1、R152x1、R152x2、7エポック事前学習済み、さらにR152x2とR200x3を14エポック事前学習済み；6つのVision Transformers、ViT-B/32、B/16、L/32、L/16、7エポック事前学習済み、さらにL/16とH/14を14エポック事前学習済み；ハイブリッドの場合、モデル名の末尾の数字はパッチサイズではなく、tにおける総ダウサンプリング比を表す。ResNetバッブーン）。

図5は、転送性能とトレーニング前の総計算量の関係である（計算コストの詳細については付録D.5を参照）。モデルごとの詳細な結果は、付録の表6に記載されている。いくつかのパターンが観察される。まず、性能/計算量のトレードオフにおいて、Vision TransformersがResNetsを支配している。ViTは、同じ性能を達成するために、約2~4倍少ない計算量を使用する（5つのデータセットの平均）。第二に、ハイブリッドは小さな計算予算ではViTをわずかに上回るが、大きなモデルではその差は消失する。この結果は、どのようなサイズでも畳み込み局所特徴処理がViTを支援すると予想されるため、やや意外である。第三に、Vision Transformersは試した範囲内では飽和しないようであり、将来のスケーリング努力の動機付けとなる。

4.5 ビジョントランスマーチャーの検査

Vision Transformerがどのように画像データを処理するかを理解し始めるために、その内部表現を分析する。Vision Transformerの第1層は、平坦化されたパッチを低次元空間に線形に投影する（式1）。図7（左）は、学習された埋め込みフィルタの上位主成分を示している。この構成要素は、各パッチ内の微細構造を低次元で表現するため、もっともらしい基底関数に似ている。

投影後、学習した位置埋め込みをパッチ表現に追加する。図7（中央）は、このモデルが画像内の距離を位置埋め込みの類似性で符号化することを学習していること、すなわち、近いパッチほど類似した位置埋め込みを持つ傾向があることを示している。さらに、行/列の構造が現れ、同じ行/列のパッチは類似した埋め込みを持つ。最後に、大きな格子では正弦波構造が見られることがある（付録D）。位置埋め込みが2次元画像のトポロジーを表現するように学習することは、手作業で2次元を意識した埋め込み変種が改善をもたらさない理由を説明する（付録D.4）。

自己アテンションにより、ViTは最下層でも画像全体の情報を統合することができる。ネットワークがこの能力をどの程度利用しているかを調査する。具体的には、注目重みに基づいて、情報が統合される画像空間における平均距離を計算する（図7右）。

この「注意距離」はCNNの受容野の大きさに類似している。我々は、いくつかの頭部が、すでに最下層にある画像のほとんどに注意を払い、情報をグローバルに統合する能力が、実際にモデルによって使用されていることを示すことを発見した。他の注意ヘッドは低レイヤーで一貫して小さな注意距離を持つ。この高度に局所化された注意は、Transformerの前にResNetを適用するハイブリッドモデルではあまり顕著ではなく（図7、右）、CNNの初期の畳み込み層と同様の機能を果たす可能性を示唆している。さらに、注意距離はネットワークの深さとともに増加する。グローバルに見ると、このモデルは分類に意味的に関連する画像領域に注目していることがわかる（図6）。

4.6 self監視

トランスマーチャーはNLPタスクで素晴らしい性能を示す。しかし、その成功の多くは、その優れたスケーラビリティだけでなく、大規模な自己教師付き事前学習（Devlin

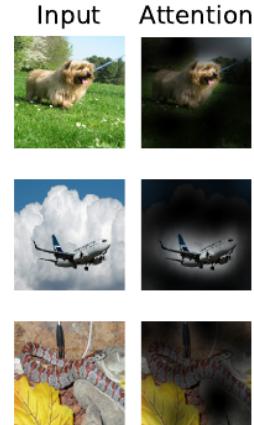


図6:出力トークンから入力空間への注意の代表例。詳細は付録D.7を参照のこと。

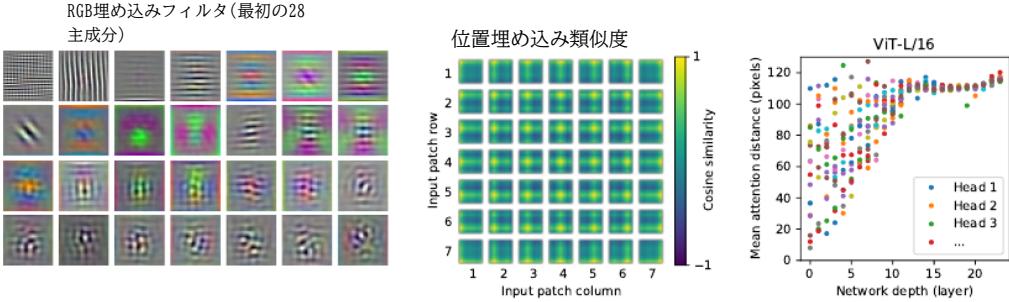


図7:左:ViT-L/32のRGB値の初期線形埋め込みフィルタ。中央: ViT-L/32の位置埋め込みの類似度。タイルは、示された行と列を持つパッチの位置埋め込みと、それ以外のパッチの位置埋め込みとの余弦類似度を示す。右: 頭部とネットワークの深さによる有人エリアの大きさ。各ドットは、1つのレイヤーにおける16個のヘッドのうちの1つについて、画像全体の平均注意距離を示す。詳細は付録D.7を参照のこと。

ら、2019;Radfordら、2018)。また、BERTで使用されているマスク言語モデリングタスクを模倣して、自己監視のためのマスクされたパッチ予測に関する予備的な探索を行う。自己教師付き事前学習により、我々の小型モデルViT-B/16はImageNetで79.9%の精度を達成し、ゼロからの学習より2%大幅に改善したが、教師付き事前学習より4%遅れている。付録B.1.2には、さらなる詳細が記載されている。対照的な事前学習(Chen et al., 2020b; He et al., 2020; Bachman et al., 2019; Hénaff et al., 2020)の探求は今後の研究に委ねる。

5 CONCLUSION

Transformersを画像認識に直接応用することを検討した。コンピュータビジョンにおける自己注意を用いた先行研究とは異なり、我々は最初のパッチ抽出ステップを除けば、画像固有の帰納的バイアスをアーキテクチャに導入しない。代わりに、我々は画像をパッチのシーケンスとして解釈し、NLPで使用されるような標準的なTransformerエンコーダによって処理する。このシンプルでスケーラブルな戦略は、大規模なデータセットでの事前学習と組み合わせることで、驚くほどうまく機能する。このように、Vision Transformerは、多くの画像分類データセットにおいて、事前学習が比較的安価でありながら、最先端技術に匹敵するか、それを上回る。

これらの初期結果は有望であるが、多くの課題が残っている。一つは、検出やセグメンテーションなど、他のコンピュータビジョントラスルにViTを適用することである。我々の結果は、Carionら(2020)の結果と相まって、このアプローチの有望性を示している。もう一つの課題は、自己教師付き事前学習法の探求を続けることである。我々の最初の実験では、自己教師付き事前学習による改善が見られたが、自己教師付き事前学習と大規模教師付き事前学習の間にはまだ大きな隔たりがある。最後に、ViTのさらなるスケーリングは、性能の向上につながる可能性が高い。

謝辞を述べる

本研究はベルリン、チューリッヒ、アムステルダムで行われた。特に、Andreas Steinerにはインフラストラクチャとコードのオープンソースリリースに多大な協力を、Joan PuigcerverとMaxim Neumannには大規模なトレーニングインフラに、Dmitry Lepikhin、Aravindh Mahendran、Daniel Keyser、Mario Lučić、Noam Shazeer、Ashish Vaswani、Colin Raffelには有益な議論をいただいた。

REFERENCES

サミラ・アブナー、ウィレム・ズイデマトランスマスターにおける注意の流れを定量化する。ACL, 2020.

フィリップ・バッハマン、R・デポン・ヒエルム、ウィリアム・ブッフウォルター。ビュー間の相互情報を最大化することで表現を学習する。2019年、NeurIPSにて。

アレクセイ・バエフスキー、マイケル・オーリニューラル言語モデリングのための適応的な入力表現。ICLR , 2019.

I. Bello, B. Zoph, Q. Le, A. Vaswani, and J. Shlens. Attention augmented convolutional networks. In ICCV, 2019.

ルーカス・バイエル、オリヴィエ・J・ヘナフ、アレクサンダー・コレスニコフ、シャオファ・ツァイ、アロン・ヴァン・デン・オード。Imagenetでできているのか?

トム・B・ブラウン、ベンジャミン・マン、ニック・ライダー、メラニー・スピア、ジャレッド・カプラン、プラフルラ・ダリワル、アルヴィン・ニーラカンタン、プラナヴ・シャム、ギリッシュ・サストリー、アマンダ・アスケル、他。言語モデルは少数ショット学習者である。arXiv, 2020。

ニコラ・カリオン、フランシスク・マサ、ガブリエル・シンネフ、ニコラ・ウスニエ、アレクサンダー・キリロフ、セルゲイ・ザゴリコ。変換器を用いたエンドツーエンドの物体検出。ECCV, 2020.

マーク・チェン、アレック・ラドフォード、リウォン・チャイルド、ジェフ・ウー、ヒーウ・ジュン。ピクセルからの生成的な事前学習ICML, 2020a。

ティン・チェン、サイモン・コーンブリス、モハマド・ノロウジ、ジェフリー・E・ヒントン。視覚表現の対比学習のためのシンプルなフレームワーク。ICML, 2020b。

陳延春、李林杰、余力成、アーメッド・エル・コリー、ファイサル・アーメッド、鎮哲、程玉、劉景景。UNITER: 普遍的な画像-テキスト表現学習。ECCV, 2020c。

リウォン・チャイルド、スコット・グレイ、アレック・ラドフォード、イリヤ・スツケバー。スペース変換器による長いシーケンスの生成。arXiv, 2019

ジャン=バティスト・コルドニエ、アンドレアス・ルーカス、マーティン・ジャッジ。自己アテンション層と畳み込み層の関係について。ICLR, 2020.

J. 鄧、董、R. ソッチャー、李、李飛飛。Imagenet: 大規模な階層型画像データベース。CVPR, 2009.

ジェイコブ・デブリン、ミンウェイ・チャン、ケントン・リー、クリスティナ・トウタノヴァ。BERT: 言語理解のための深い双方向変換器の事前学習。2019年NAACLにて。

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvan Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On robustness and transferability of convolutional neural networks. arXiv, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

何凱明、範浩基、吳玉信、謝彩寧、ロス・ガーシック。運動量対比教師なし視覚表現学習。CVPR, 2020.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. arXiv, 2019.

胡漢、顧建源、張正、戴吉峰、ウェイ・イーチェン。物体検出のための関係ネットワークCVPR, 2018.

ハン・フー、チャン・ジェン、謝振達、林峻。画像認識のための局所関係ネットワークICCV, 2019.

黃紫龍、王興剛、ウェイ・ユンチャオ、黃麗超、シー・フンフリー、劉文字、黃トーマス S.。Cnet: 意味分割のための十字注意。ICCV, 2020.

オリヴィエ・J・ヘナフ、アラビンド・スリニヴァス、ジェフリー・デ・フォウ、アリ・ラザヴィ、カール・ドーシュ、S・M・アリ・エスラミ、アーロン・ヴァン・デン・オード。対照的予測符号化によるデータ効率の良い画像認識。ICML, 2020.

セルゲイ・イオフェ、クリスチャン・セゲディバッチ正規化：内部共変量シフトを減らすことでディープネットワークの学習を高速化する。2015。

ディエドリク・P・キングマ、ジミー・パAdam：確率的最適化のための手法。ICLR, 2015.

アレクサンダー・コレスニコフ、ルーカス・ペイヤー、シャオファ・ツァイ、ジョン・ブイグセルバー、ジェシカ・ヨン、シルヴァン・グリー、ニール・ホールズビー。ビッグトランスファー(BiT)：一般的な視覚表現学習。ECCV, 2020.

Alex Krizhevsky. 小さな画像から多層の特徴を学習する。技術報告書、2009年

アレックス・クリシェフスキ、イリヤ・スツケバー、ジェフリー・E・ヒントン。深層畳み込みニューラルネットワークによるイメージネット分類。NIPS, 2012.

Yルクン、B.ボザー、J.デンカー、D.ヘンダーソン、R.ハワード、W.ハバード、L.ジャッケル。手書き郵便番号認識に適用されるバックプロパゲーション。神經計算, 1:541–551, 1989.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv*, 2020.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A Simple and Performant Baseline for Vision and Language. In *Arxiv*, 2019.

フランチエスコ・ロカテロ、ディルク・ヴァイセンボーン、トマス・ウンターティナー、アラビンド・マヘンドラ、ゲオルク・ハイゴルド、ヤコブ・ウスコレイト、アレクセイ・ドソヴィツキー、トマス・キップ。スロットアテンションによるオブジェクト中心学習arXiv, 2020。

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In *NeurIPS*. 2019.

ダルヴ・マハジャン、ロス・ガーシック、ヴィニュシュ・ラマナサン、何凱明、マノハル・パルリ、イクスアン・リー、アシュウイン・バランベ、ローレンス・ファン・デル・マーテン。弱教師付き事前トレーニングの限界を探る。2018年ECCVにて。

M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

ニキ・パルマー、アシシュ・ヴァスワニ、ヤコブ・ウスコレイト、ルカシュ・カイザー、ノーム・シャゼール、アレクサンダー・クー、ダステイン・トラン。画像変換器。ICML, 2018 にて。

B. T. Polyak and A. B. Juditsky. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992. doi: 10.1137/0330046. URL <https://doi.org/10.1137/0330046>.

Siyuan Qiao, Huiyu Wang, Chenxi Liu, Wei Shen, and Alan Yuille. Weight standardization. *arXiv preprint arXiv:1903.10520*, 2019.

アレック・ラドフォード、カルティク・ナラシマン、ティム・サリマンス、イリヤ・スツケバー。教師なし学習による言語理解の向上テクニカルレポート、2018年

アレック・ラドフォード、ジェフ・ウー、リウォン・チャイルド、デビッド・ルアン、ダリオ・アモディ、イリヤ・スツケバー。言語モデルは教師なしマルチタスク学習者である。テクニカルレポート、2019年

プラジット・ラマチャンドラン、ニキ・パルマー、アシシュ・ヴァスワニ、イルワン・ベロ、アンセルム・レフスカヤ、ジョン・シュレンズ。視覚モデルにおける単独の自己注意。2019年、NeurIPSにて。

チェン・サン、アビナブ・シュリバスター、サウラブ・シン、アビナブ・グプタ。ディープラーニング時代のデータの理不尽な有効性を再検討する。ICCV, 2017.

チェン・サン、オースティン・マイヤーズ、カール・ボンドリック、ケビン・マーフィー、コーデリア・シュミッド。Videobert: ビデオと言語表現学習のためのジョイントモデル。ICCV, 2019.

ユーゴ・トゥブロン、アンドレア・ヴェダルディ、マタイス・ドウーズ、ヘルヴェ・ジェグー。訓練とテストの解像度の不一致を修正する。NeurIPS, 2019.

ユーゴ・トゥブロン、アンドレア・ヴェダルディ、マタイス・ドウーズ、ヘルヴェ・ジェグー。訓練とテストの解像度の不一致を修正する: Fixefficientnet. arXivプレプリントarXiv:2003.08237, 2020.

Michael Tschannen, Josip Djolonga, Marvin Ritter, Aravindh Mahendran, Neil Houlsby, Sylvain Gelly, and Mario Lucic. Self-supervised learning of video-induced visual invariances. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020a.

Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. *arXiv preprint arXiv:2003.07853*, 2020b.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *ACL*, 2019.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. In *ICLR*, 2019.

Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arxiv*, 2020.

Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. In *CVPR*, 2020.

蔡曉華、アビタル・オリバー、アレクサンダー・コレスニコフ、ルーカス・ベイヤー。S⁴ L: 自己教師付き半教師付き学習. ICCV, 2019a.

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019b.

Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020.

Models	Dataset	Epochs	Base LR	LR decay	Weight decay	Dropout
ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/32	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-H/14	JFT-300M	14	$3 \cdot 10^{-4}$	linear	0.1	0.0
R50x{1,2}	JFT-300M	7	10^{-3}	linear	0.1	0.0
R101x1	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R152x{1,2}	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/32	JFT-300M	7	$2 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-B/{16,32}	ImageNet-21k	90	10^{-3}	linear	0.03	0.1
ViT-L/{16,32}	ImageNet-21k	30/90	10^{-3}	linear	0.03	0.1
ViT-*	ImageNet	300	$3 \cdot 10^{-3}$	cosine	0.3	0.1

表3:学習用ハイパーパラメータすべてのモデルはバッチサイズ4096、学習率ウォームアップ10kステップで学習される。ImageNetでは、グローバルノルム1で勾配クリッピングを追加適用することが有益であることがわかった。学習分解能は224である。

APPENDIX

マルチヘッド・エルフ・アテンション

標準的なqkv自己注意(SA, Vaswani et al. (2017))は、ニューラルアーキテクチャのための一般的なビルディングブロックである。入力シーケンス $\mathbf{z} \in \mathbb{R}^{N \times D}$ の各要素について、シーケンス内の全ての値 v に対する重み付き和を計算する。注目重み A_{ij} は、シーケンスの2つの要素と、それぞれのクエリ q^i とキー k^j 表現との間のペアワイズ類似度に基づいている。

$$[\mathbf{q}, \mathbf{k}, \mathbf{v}] = \mathbf{z} \mathbf{U}_{qkv} \quad \mathbf{U}_{qkv} \in \mathbb{R}^{D \times 3D_h}, \quad (5)$$

$$A = \text{softmax} \left(\mathbf{q} \mathbf{k}^\top / \sqrt{D_h} \right) \quad A \in \mathbb{R}^{N \times N}, \quad (6)$$

$$\text{SA}(\mathbf{z}) = A \mathbf{v}. \quad (7)$$

マルチヘッド自己注意(MSA)はSAの拡張で、「ヘッド」と呼ばれる k 個の自己注意操作を並列に実行し、それらの連結出力を投影する。 k を変化させても計算量とパラメータ数を一定に保つため、 D_h (式5)は通常 D/k に設定される。

$$\text{MSA}(\mathbf{z}) = [\text{SA}_1(z); \text{SA}_2(z); \dots; \text{SA}_k(z)] \mathbf{U}_{msa} \quad \mathbf{U}_{msa} \in \mathbb{R}^{k \cdot D_h \times D} \quad (8)$$

B 実験の詳細

B.1 TRAINING

表3は、我々の異なるモデルのトレーニングセットアップをまとめたものである。ImageNetでゼロからモデルを学習する場合、強い正則化が鍵となることがわかった。ドロップアウトを使用する場合、qkv-projectionを除くすべての密な層の後に適用され、パッチ埋め込みに位置情報を追加した後に直接適用される。ハイブリッドモデルは、ViTモデルと全く同じ設定で学習される。最後に、すべてのトレーニングは解像度224で行われる。

B.1.1 FINE-TUNING

全てのViTモデルを0.9の運動量でSGDを用いて微調整する。表4の学習率の範囲を参照し、学習率に対して小さなグリッド探索を実行する。そのために、学習セット(Pets and Flowersは10%、CIFARは2%、ImageNetは1%)の小さなサブスプリットを開発セットとして使用し、残りのデータで学習する。最終的な結果については、トレーニングセット全体でトレーニングを行い、それぞれのテストデータで評価する。ResNetsとハイブリッドモデルの微調整には、全く同じ設定を使用するが、ImageNetだけは例外で、学習率掃引に0.06を追加する。

Dataset	Steps	Base LR
ImageNet	20 000	{0.003, 0.01, 0.03, 0.06}
CIFAR100	10 000	{0.001, 0.003, 0.01, 0.0}
CIFAR10	10 000	3} {0.001, 0.003, 0.01, 0.
Oxford-IIIT Pets	500	.03} {0.001, 0.003, 0.01,
Oxford Flowers-102	500	0.03}
VTAB (19 tasks)	2 500	0.01

表4:微調整のためのハイパーパラメータ。すべてのモデルは、コサイン学習率減衰、バッチサイズ512、重み減衰なし、グローバルノルム1でのgradクリッピングで微調整される。特に言及がない場合、微調整の解像度は384である。

さらに、ResNetsについては、Kolesnikovら(2020)のセットアップも実行し、この実行と我々の掃引で最良の結果を選択する。最後に、特に言及がない場合、すべての微調整実験は384の解像度で実行される(トレーニングとは異なる解像度で微調整を実行するのが一般的である(Kolesnikov et al.)

ViTモデルを別のデータセットに転送する際、ヘッド全体(2つの線形層)を削除し、ターゲットデータセットが必要とするクラス数を出力する、初期化ゼロの線形層で置き換える。これは、単に最後のレイヤーを再初期化するよりも、少しロバストであることがわかった。

VTABについては、Kolesnikovら(2020)のプロトコルに従い、全てのタスクで同じハイパーパラメータ設定を使用する。学習率0.01を使用し、2500ステップの学習を行う(表4)。我々は、2つの学習率と2つのスケジュールに対して小さなスイープを実行し、200例の検証セットで最も高いVTABスコアを持つ設定を選択することで、この設定を選択した。タスク固有の入力解像度を使用しない以外は、Kolesnikov et al. (2020)で使用された前処理に従う。代わりに、Vision Transformerはすべてのタスクで高解像度(384×384)から最も恩恵を受けることがわかった。

B.1.2 自己監視

我々は予備的な自己監視実験のために、マスクされたパッチ予測目的を採用する。そのために、50%のパッチ埋め込みを、学習可能な[マスク]埋め込み(80%)、ランダムな他のパッチ埋め込み(10%)、またはそのまま(10%)に置き換えることで破損させる。この設定は、Devlinら(2019)が言語に使用したものと非常に似ている。最後に、それぞれのパッチ表現を用いて、すべての破損パッチの3ビット、平均色(つまり合計512色)を予測する。

自己教師付きモデルを1Mステップ学習させた(約14エポック)、バッチサイズ4096、JFT。Adamを使用し、基本学習率は $2 - 10^{-4}$ 、ウォームアップは10kステップ、コサイン学習率減衰とする。事前学習の予測対象として、以下の設定を試した: 1) 平均、3ビットの色のみを予測する(すなわち、512色の1予測)、2) 3ビットの色を持つ 16×16 パッチの 4×4 縮小版を並列に予測する(すなわち、512色の16予測)、3) L2 を用いたフルパッチへの回帰(すなわち、3つのRGBチャンネルに対する256回帰)。意外なことに、L2はやや悪化したものの、すべてがかなりうまく機能することがわかった。オプション1)については、数発で最高のパフォーマンスを示したため、最終結果のみを報告する。Devlinら(2019)が使用した15%の破損率でも実験を行ったが、結果は我々の数ショットメトリクスでも若干悪化した。

最後に、マスクされたパッチ予測のインスタンス化は、ImageNet分類で同様の性能向上をもたらすために、このような膨大な事前学習やJFTのような大規模なデータセットを必要としないことを述べておきたい。すなわち、100k回の事前学習ステップの後、ダウンストリームの性能に対する収穫が遞減することが観察され、ImageNetで事前学習した場合にも同様の効果が見られる。

追加結果について

論文で紹介した図に対応する詳細な結果を報告する。表5は論文の図3に対応し、サイズが大きくなるデータセットで事前学習した様々なViTモデルの転送性能を示している: ImageNet、ImageNet-21k、JFT-300M。表6は以下に対応する。

		ViT-B/16	ViT-B/32	ViT-L/16	ViT-L/32	ViT-H/14
ImageNet	CIFAR-10	98.13	97.77	97.86	97.94	-
	CIFAR-100	87.13	86.31	86.35	87.07	-
	ImageNet	77.91	73.38	76.53	71.16	-
	ImageNet ReAL	83.57	79.56	82.19	77.83	-
	Oxford Flowers-102	89.49	85.43	89.66	86.36	-
	Oxford-IIIT-Pets	93.81	92.04	93.64	91.35	-
ImageNet-21k	CIFAR-10	98.95	98.79	99.16	99.13	99.27
	CIFAR-100	91.67	91.97	93.44	93.04	93.82
	ImageNet	83.97	81.28	85.15	80.99	85.13
	ImageNet ReAL	88.35	86.63	88.40	85.65	88.70
	Oxford Flowers-102	99.38	99.11	99.61	99.19	99.51
	Oxford-IIIT-Pets	94.43	93.02	94.73	93.09	94.82
JFT-300M	CIFAR-10	99.00	98.61	99.38	99.19	99.50
	CIFAR-100	91.87	90.49	94.04	92.52	94.55
	ImageNet	84.15	80.73	87.12	84.37	88.04
	ImageNet ReAL	88.85	86.27	89.99	88.28	90.33
	Oxford Flowers-102	99.56	99.27	99.56	99.45	99.68
	Oxford-IIIT-Pets	95.80	93.40	97.11	95.83	97.56

表5: ImageNet、ImageNet-21k、JFT300Mで事前学習した場合の、様々なデータセットにおけるVision Transformerのトップ1精度(単位:%)。これらの値は本文の図3に対応する。モデルは384の解像度で微調整される。なお、ImageNetの結果は、表2の結果を得るために使用した追加技術(Polyak平均化、512解像度画像)なしで計算されている。

name	Epochs	ImageNet	ImageNet ReAL	CIFAR-10	CIFAR-100	Pets	Flowers	exaFLOPs
ViT-B/32	7	80.73	86.27	98.61	90.49	93.40	99.27	55
ViT-B/16	7	84.15	88.85	99.00	91.87	95.80	99.56	224
ViT-L/32	7	84.37	88.28	99.19	92.52	95.83	99.45	196
ViT-L/16	7	86.30	89.43	99.38	93.46	96.81	99.66	783
ViT-L/16	14	87.12	89.99	99.38	94.04	97.11	99.56	1567
ViT-H/14	14	88.08	90.36	99.50	94.71	97.11	99.71	4262
ResNet50x1	7	77.54	84.56	97.67	86.07	91.11	94.26	50
ResNet50x2	7	82.12	87.94	98.29	89.20	93.43	97.02	199
ResNet101x1	7	80.67	87.07	98.48	89.17	94.08	95.95	96
ResNet152x1	7	81.88	87.96	98.82	90.22	94.17	96.94	141
ResNet152x2	7	84.97	89.69	99.06	92.05	95.37	98.62	563
ResNet152x2	14	85.56	89.89	99.24	91.92	95.75	98.75	1126
ResNet200x3	14	87.22	90.15	99.34	93.53	96.32	99.04	3306
R50x1+ViT-B/32	7	84.90	89.15	99.01	92.24	95.75	99.46	106
R50x1+ViT-B/16	7	85.58	89.65	99.14	92.63	96.65	99.40	274
R50x1+ViT-L/32	7	85.68	89.04	99.24	92.93	96.97	99.43	246
R50x1+ViT-L/16	7	86.60	89.72	99.18	93.64	97.03	99.40	859
R50x1+ViT-L/16	14	87.12	89.76	99.31	93.89	97.36	99.11	1668

表6: モデルのスケーリング実験の詳細結果。これらは本論文の図5に対応する。いくつかのデータセットにおける転送精度と、事前学習計算量(exaFLOPs)を示す。

図5は論文から引用したもので、サイズの異なるViT、ResNet、ハイブリッドモデルの転送性能と、それらの事前学習による推定計算コストを示している。

d 追加分析

D.1 SGD vs. RESネットのためのダム

ResNetsは通常SGDで学習されるが、Adamをオプティマイザとして使用するのは型破りである。ここでは、この選択の動機となった実験を示す。

Dataset	ResNet50		ResNet152x2	
	Adam	SGD	Adam	SGD
ImageNet	77.54	78.24	84.97	84.37
CIFAR10	97.67	97.46	99.06	99.07
CIFAR100	86.07	85.17	92.05	91.06
Oxford-IIIT Pets	91.11	91.00	95.37	94.79
Oxford Flowers-102	94.26	92.06	98.62	99.32
Average	89.33	88.79	94.01	93.72

表7: AdamとSGDで事前学習したResNetモデルの微調整。

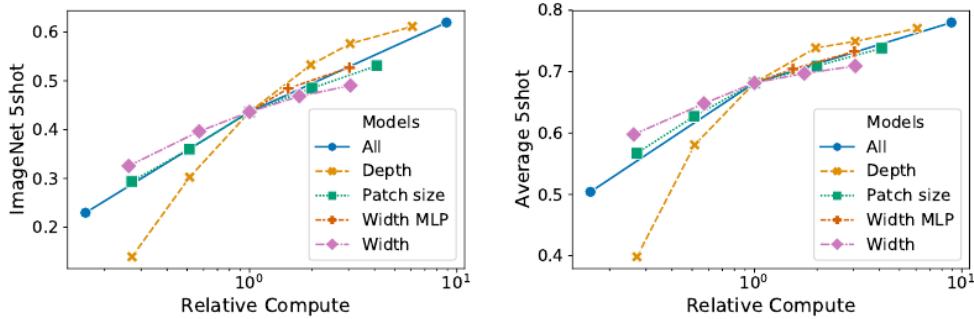


図8:Vision Transformerの異なるモデル次元のスケーリング。

すなわち、SGDとAdamを用いてJFTで事前学習した50x1と152x2の2つのResNetの微調整性能を比較する。SGDには、Kolesnikovら(2020)が推奨するハイパーパラメータを使用する。結果を表7に示す。Adamの事前学習は、ほとんどのデータセットと平均でSGDの事前学習を上回る。これは、JFT上でResNetsを事前学習するために使用するオプティマイザとしてAdamを選択することを正当化する。なお、絶対値はKolesnikovら(2020)の報告値より低く、これは30エポックではなく7エポックのみ事前学習を行っているためである。

D.2 トランスフォーマー形状

Transformerアーキテクチャの異なる次元のスケーリングについてアブレーションを実行し、非常に大きなモデルへのスケーリングに最適なものを見つめた。図8は、ImageNetにおける5ショットの性能を、異なる構成で示したものである。すべての構成は、 $D = 1024$, $D_{MLP} = 2048$ 、パッチサイズ32、全線の交点の8層のViTモデルに基づいている。深さをスケーリングすると、64層まではっきりと見える最大の改善が得られることがわかる。しかし、16階層以降にはすでに収穫過増が見られる。興味深いことに、ネットワークの幅を広げると、変化が最も小さくなるようである。パッチサイズを小さくし、有効配列長を長くすることで、パラメータを導入することなく、驚くほどロバストな改善を示す。これらの結果は、計算がパラメータ数よりも性能の予測因子として優れている可能性があり、スケーリングがあれば幅よりも深さを重視すべきであることを示唆している。全体として、すべての次元を比例的にスケーリングすることで、ロバストな改善が得られることがわかった。

D.3 ID.3 ヘッドタイプとCLASトークン

オリジナルのTransformerモデルにできるだけ近づけるため、画像表現として追加の[cls]トークンを使用した。このトークンの出力は、tanhを非線形とする小さな多層パーセプトロン(MLP)を介して、単一隠れ層のクラス予測に変換される。

この設計はテキストのTransformerモデルから継承されており、本論文ではこれを用いる。画像パッチ埋め込みのみを使用し、グローバル平均プーリング(GAP)を行い、その後、ResNetの最終特徴マップと同様に線形分類器を使用した最初の試みは、非常に悪い結果となった。

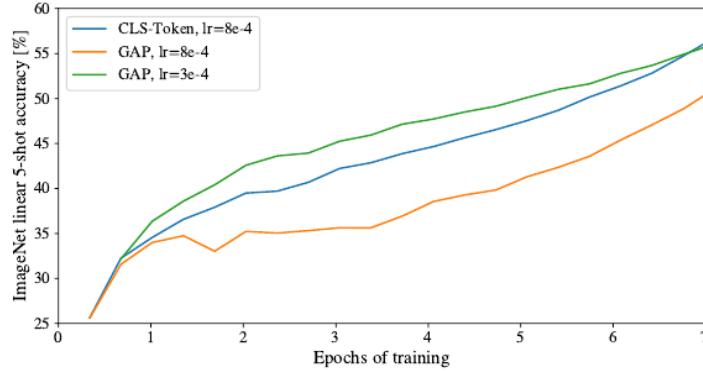


図9: クラストークン分類器とグローバル平均プーリング分類器の比較。両者は同じように機能するが、異なる学習率を必要とする。

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

表8:ViT-B/16モデルによる位置埋め込みをImageNet 5-shot linearで評価したアブレーション研究の結果。

しかし、これは余分なトークンによるものでも、GAP操作によるものでもないことがわかった。その代わりに、性能の違いは、異なる学習率の要件によって完全に説明される、図9を参照。

D.4 オプションの埋め込み

位置埋め込みを用いた空間情報の符号化方法の違いによるアブレーションを行った。以下のケースを試した：

- 位置情報を提供しない：入力をパッチの袋として考える。
- 1次元の位置埋め込み。入力をラスター順のパッチ列とみなす(本論文の他のすべての実験ではデフォルト)。
- 2次元の位置埋め込み：入力を2次元のパッチのグリッドとして考える。この場合、X-embeddingとY-embeddingの2組の埋め込みが学習され、それぞれ1軸分のサイズはD/2である。そして、入力のパス上の座標に基づいて、XとYの埋め込みを連結し、そのパッチの最終的な位置埋め込みを得る。
- 相対位置埋め込み：相対位置埋め込み：絶対位置の代わりに空間情報をエンコードするために、パッチ間の相対距離を考慮する。そのために、1次元の相対的注意(Relative Attention)を使用し、可能なすべてのパッチのペアの相対距離を定義する。したがって、与えられたペア(1つはクエリ、もう1つは注目メカニズムにおけるキー/値)ごとに、オフセット $p_q - p_k$ があり、各オフセットは埋め込みと関連付けられる。次に、元のクエリ(クエリの内容)を使用し、相対位置埋め込みをキーとして使用する、単純に余分な注意を実行する。次に、相対的注意のロジットをバイアス項として使用し、ソフトマックスを適用する前に、主注意(内容ベースの注意)のロジットに追加する。

空間情報を符号化するさまざまな方法に加えて、この情報をモデルに組み込むさまざまな方法も試した。1次元と2次元の位置埋め込みについては、3つの異なるケースを試した：(1)それらのシステムモデルの

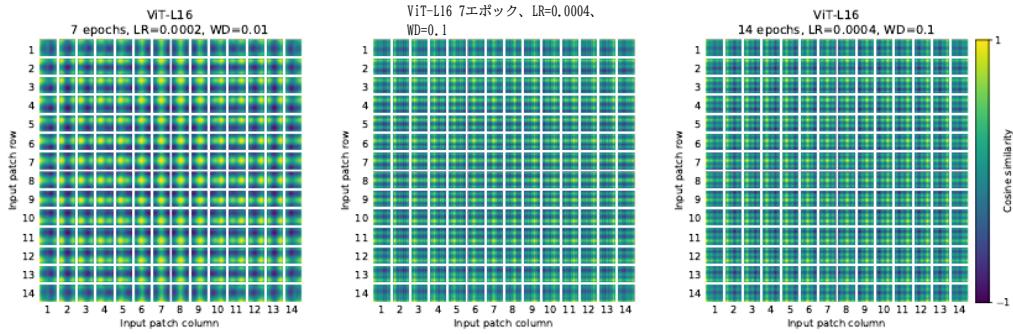


図10:異なるハイパーパラメータで学習したモデルの位置埋め込み。

直後とTransformerエンコーダに入力を与える前に、入力に位置埋め込みを追加する(本論文の他のすべての実験にわたるデフォルト)、(2)各層の最初に位置埋め込みを学習して入力に追加する、(3)各層の最初に(層間で共有)入力に学習した位置埋め込みを追加する。

表8は、ViT-B/16モデルを用いたアブレーション研究の結果をまとめたものである。このように、位置埋め込みを行わないモデルと位置埋め込みを行うモデルの性能には大きな差があるものの、位置情報の符号化方法の違いによる差はほとんどないことがわかる。我々のTransformerエンコーダはピクセルレベルではなくパッチレベルの入力で動作するため、空間情報のエンコード方法の違いはあまり重要ではないと推測される。より正確には、パッチレベルの入力では、空間次元は元のピクセルレベルの入力よりもはるかに小さく、例えば、 224×224 に対して 14×14 であり、この解像度で空間関係を表現する学習は、これらの異なる位置エンコーディング戦略にとって同様に容易である。それでも、ネットワークが学習する位置埋め込み類似度の具体的なパターンは、学習ハイパーパラメータに依存する(図10)。

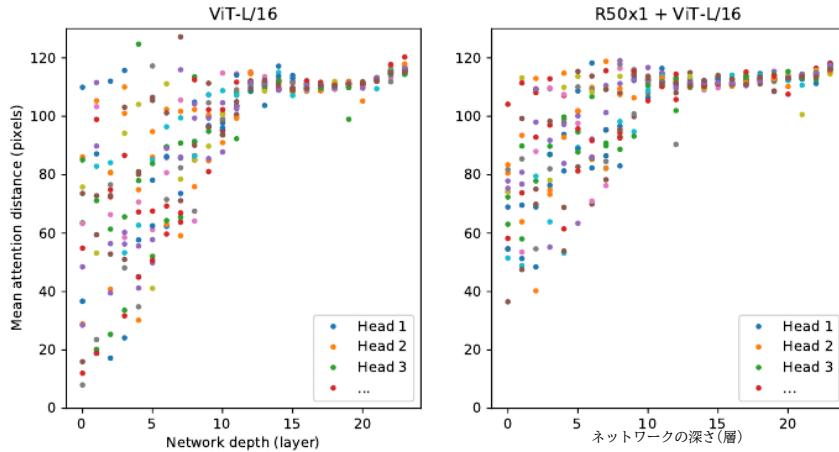


図11:頭部とネットワークの深さによる有人エリアの大きさ。注意距離は、128枚の画像例について、クエリ画素と他の全ての画素との距離を、注意重みで重み付けして平均化することにより計算した。各ドットは、1つのレイヤーにおける16個のヘッドのうちの1つについて、画像全体の平均注意距離を示す。画像幅は224ピクセル。

D.5 経験的計算コスト

また、車線幅やキャッシュサイズなどの詳細により、理論的なFLOPsでは必ずしも予測できない、我々のハードウェア上でのアーキテクチャの実世界の速度にも興味がある。

この目的のために、TPUv3アクセラレータ上で、関心のある主要なモデルの推論速度のタイミングを実行する。推論速度とバックプロップ速度の差は、モデルに依存しない一定の係数である。

図12(左)は、様々な入力サイズにおいて、1つのコアが1秒間にいくつの画像を扱うことができるかを示している。すべてのポイントは、広い範囲のバッチサイズにわたって測定されたピーク性能を指す。見てわかるように、画像サイズによるViTの理論的な2次スケーリングは、最大の解像度で最大のモデルに対してかろうじて起こり始めるだけである。

もう一つの注目すべき量は、各モデルがコアに収まる最大のバッチサイズであり、大きなデータセットへのスケーリングには大きい方が良い。図12(右)は、同じモデルセットについて、この量を示している。これは、大規模なViTモデルがResNetモデルよりもメモリ効率の面で明らかに有利であることを示している。

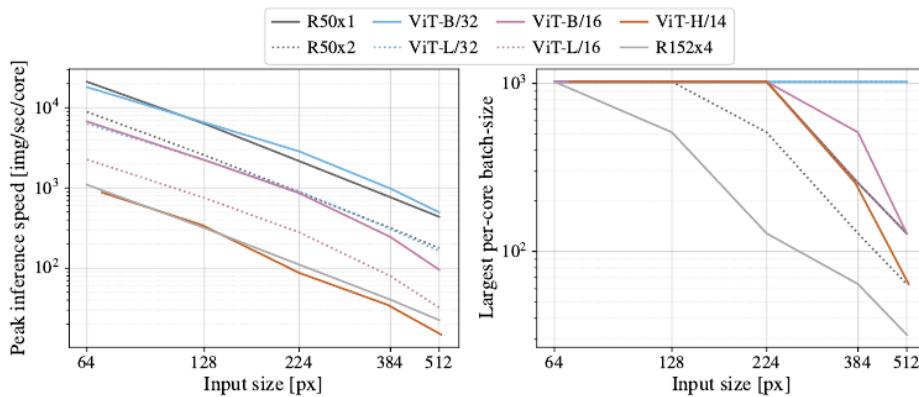


図12:左:入力サイズに渡る様々なアーキテクチャの実際のウォールクロックタイミング。ViTモデルは類似のResNetsに匹敵する速度を持つ。右:右:入力サイズに関係なく、様々なアーキテクチャを持つデバイスでコアあたり最大のバッチサイズフィッティング。ViTモデルの方が明らかにメモリ効率が良い。

D.6 軸方向の注意

軸方向注意(Huang et al., 2020; Ho et al., 2019)は、多次元テンソルとして組織化された大きな入力に対して自己注意を実行する、シンプルでありながら効果的な手法である。軸方向注意の一般的な考え方とは、入力の平坦化されたバージョンに1次元の注意を適用する代わりに、それぞれが入力テンソルの1つの軸に沿って、複数の注意操作を実行することである。軸方向注意では、各注意は特定の軸に沿った情報を混合し、他の軸に沿った情報は独立に保つ。この線に沿って、Wangら(2020b)は、ResNet50のカーネルサイズ3×3の畠み込みをすべて、相対位置エンコーディングによって補強された軸方向の自己注意、すなわち行と列の注意に置き換えるAxialResNetモデルを提案した。AxialResNetをベースラインモデルとして実装した³。

さらに、ViTを1次元のパッチ列ではなく、2次元の形状の入力を処理するように修正し、軸変換ブロックを組み込んだ。このブロックでは、自己注意の後にMLPが続く代わりに、行自己注意+MLPの後に列自己注意+MLPが続く。

図13は、ImageNet 5shot linearにおけるAxial ResNet、Axial-ViT-B/32、Axial-ViT-B/16の性能を、FLOP数と推論時間(1秒あたりの例)の両方で、事前学習計算と比較して示したものである。Axial-ViT-B/32とAxial-ViT-B/16は、性能的にはViT-Bよりも優れているが、計算量が多くなるという代償があることがわかる。

我々の実装は、<https://github.com/csrhddlam/axial-deeplab>にあるオープンソースのPyTorch実装に基づいている。我々の実験では、(Wang et al., 2020b)で報告されたスコアを精度の点で再現したが、我々の実装はオープンソースの実装と同様に、TPU上で非常に遅い。そのため、大規模な実験に使用することができなかった。これらは、慎重に最適化された実装によって解き放つことができる。

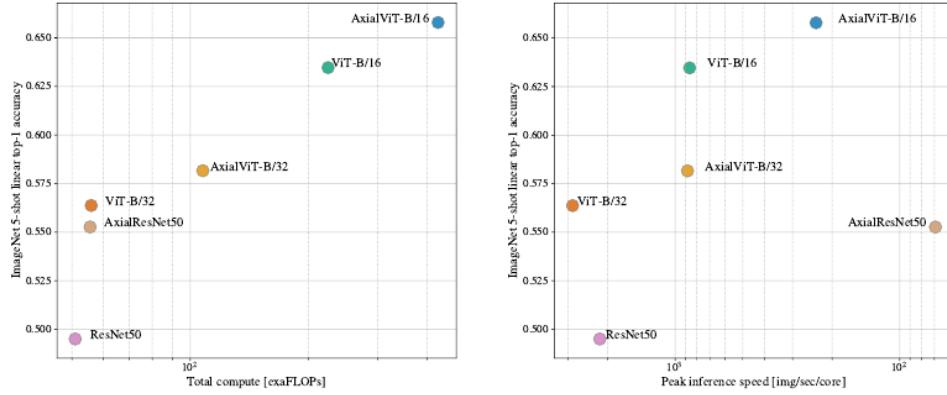


図13:ImageNet 5-shot linearにおけるAxial-Attentionベースのモデルの性能(トップ1精度)と、FLOP数(左)および推論時間(左)における速度。

これは、Axial-ViTモデルにおいて、グローバルな自己注意を持つ各Transformerブロックは、行と列の自己注意を持つ2つのAxial Transformerブロックに置き換えられ、自己注意が動作するシーケンス長はアキシャルの場合の方が小さいが、Axial-ViTブロックごとに余分なMLPが存在するためである。AxialResNetでは、精度/計算量のトレードオフの点では妥当に見えるが(図13左)、TPU上では素朴な実装が非常に遅い(図13右)。

D.7 注意の分散

ViTが自己注意を用いて画像全体の情報を統合する方法を理解するために、異なる層における注意の重みがまたがる平均距離を分析した(図11)。この「注意距離」はCNNの受容野の大きさに類似している。平均注意距離は下位レイヤーのヘッド間で大きく変動し、あるヘッドは画像の大部分に注意を向け、他のヘッドはクエリ位置またはその近傍の小さな領域に注意を向ける。深さが増すにつれて、注意距離はすべての頭部で増加する。ネットワークの後半では、ほとんどのヘッドがトークンに広くアテンションしている。

D.8 注意マップ

出力トークンから入力空間への注意のマップを計算するために(図6と14)、Attention Rollout (Abnar & Zuidema, 2020)を使用した。簡単に説明すると、ViT-L/16の注目重みを全頭で平均化し、全層の重み行列を再帰的に掛け合わせた。これは、すべての層を通してトークン間の注意の混合を説明する。

D.9 対象物のネット結果

また、Kolesnikov et al. (2020)の評価設定に従い、ObjectNetベンチマークで我々のフラッグシップモデルViT-H/14を評価した結果、トップ5精度82.1%、トップ1精度61.7%となった。

D.10 VTAB Bリーアクダウ

表9は、VTAB-1kの各タスクで達成されたスコアである。

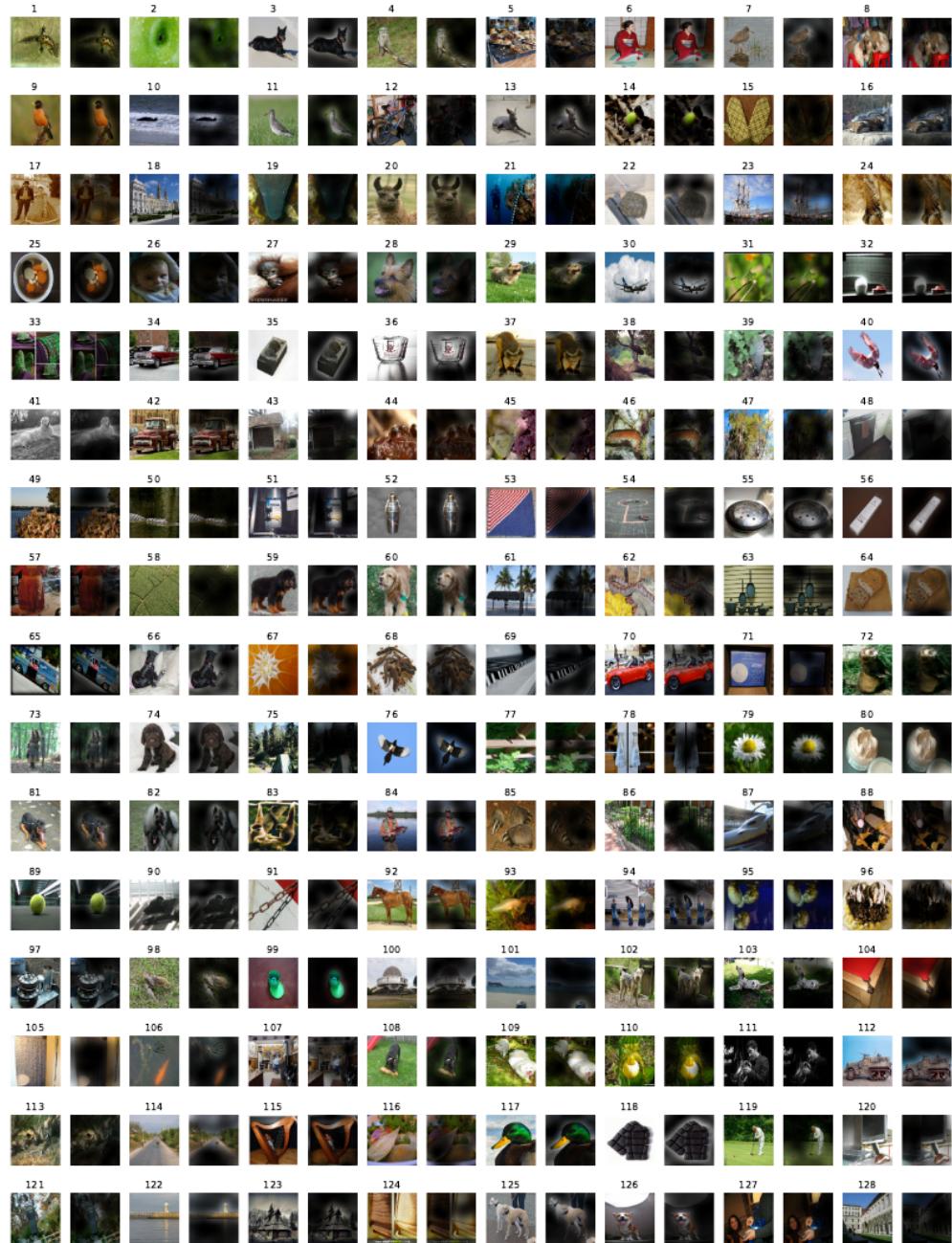


図14:図6(ランダム選択)と同様のアテンションマップのさらなる例。

表9: タスク間のVTAB-1k性能の内訳。

	Caltech101	CIFAR-100	DTD	Flowers102	Pets	Sun397	SVHN	Camelyon	EuroSAT	ReSisc45	Retinopathy	Clevr-Count	Clevr-Dist	DMLab	dSpr-Loc	dSpr-Oni	KITTI-Dist	sNORB-Azim	sNORB-Elev	Mean
ViT-H/14 (JFT)	95.3	85.5	75.2	99.7	97.2	65.0	88.9	83.3	96.7	91.4	76.6	91.7	63.8	53.1	79.4	63.3	84.5	33.2	51.2	77.6
ViT-L/16 (JFT)	95.4	81.9	74.3	99.7	96.7	63.5	87.4	83.6	96.5	89.7	77.1	86.4	63.1	49.7	74.5	60.5	82.2	36.2	51.1	76.3
ViT-L/16 (I21k)	90.8	84.1	74.1	99.3	92.7	61.0	80.9	82.5	95.6	85.2	75.3	70.3	56.1	41.9	74.7	64.9	79.9	30.5	41.7	72.7